

COMPLIED OBJECTIVE REFERENTIAL CONSTRAINTS IN A RELATIONAL DATABASE HAVING DUAL CHAIN RELATIONSHIP DESCRIPTORS LINKED IN DATA RECORD TABLES

This application is a continuation of application Ser. No. 249,049, filed on Sep. 23, 1988, now abandoned.

BACKGROUND OF THE INVENTION

1. Technical Field

This invention relates to relational data base management systems, and more particularly to the structural representation of referential constraints within the data base manager.

2. Description of the Prior Art

A data base management system is a computer system for recording and maintaining data. In a relational data base management system, data is stored in "tables" which can be viewed as having horizontal rows and vertical columns. The Database 2 product of the International Business Machines Corporation (IBM) is an example of a typical relational data base management system.

Within relational data bases, an important function is that of "referential integrity". Referential integrity ensures the consistency of data values between related columns of two different tables (or of the same table) by enforcing required relationships between tables' columns. These required relationships are known as "referential constraints". A row in a "dependent table" possesses referential integrity with respect to a constraint if the value of its "foreign key" matches the value of a "primary key" in some row of a "parent table", or if the value of its foreign key is null, i.e. which contains no value. In other words, every row in the dependent table which has a non-null value must have a corresponding parent row in the parent table. If a dependent row's foreign key has no matching primary key value in the parent table, then that referential constraint is violated and there is a loss of referential integrity in the data base comprising those tables. To enforce referential constraints and thereby maintain the data base's referential integrity, the system must ensure that non-null foreign key values always have corresponding primary key values. In implementations of referential integrity the system also ensures that primary key values are unique, a property known as "entity integrity".

By way of example, consider an EMPLOYEE table that contains employee and department numbers, and a DEPARTMENT table that contains department numbers. Referential integrity might require that for every department number in the EMPLOYEE table there must be an equal and unique department number in the DEPARTMENT table. This would require a referential constraint defined on the EMPLOYEE table. The department number in the DEPARTMENT table would be the primary key, and the department number of the EMPLOYEE table would be the foreign key, in this constraint.

Referential constraints must be enforced whenever the data of a data base is manipulated so as to affect primary or foreign keys. In relational data base management systems which use the Structured Query Language (SQL), data is primarily modified by the LOAD, INSERT, DELETE, and UPDATE commands and their resulting operations. The LOAD and INSERT

commands both add (insert) data to the data base, with LOAD typically adding many rows and INSERT adding only a few. DELETE deletes one or more rows, and UPDATE changes the contents of one or more rows. Whenever one of these operations occurs, the referential constraints involving the modified rows must be enforced to ensure the data base's referential integrity.

One method of maintaining referential integrity in a relational data base management system provides the system with means for supporting procedures (programs or routines) residing outside the system which are executed when certain predefined events occur. An example of such a procedure would be to execute a particular program whenever data is inserted into a particular table. The procedure might update an index on the table, or enforce a referential constraint on the newly inserted data. This latter would be an example of a "procedural" implementation of referential integrity. Several relational data base management products have added procedural implementations of referential integrity.

Procedural implementations of referential integrity suffer from several drawbacks which make them slow and inefficient. Because the procedures are external (outside the system), they require extra processing at the interface between the system and the procedure. This processing overhead is not incurred by internal subsystems within the overall system. There is thus a need for an implementation of referential integrity which does not incur the processing overhead associated with external procedures.

More importantly, because external procedures are invoked before or after (but not while) the system modifies the data, the data must be accessed twice—once by the system and again by the procedure. This doubling of the number of data accesses can greatly reduce the system's overall speed. There is thus a need also for an implementation of referential integrity which accesses newly modified data only once, eliminating the redundant double access associated with procedural implementations.

Procedural implementations of referential integrity have yet another disadvantage—the constraints they implement are comprehensible only to computer programmers. The programming languages used to write the procedures are seldom understandable to the data base user, and the process of changing the constraint is impossible for the ordinary user of the data base. There is a need for an implementation of referential integrity which allows non-programmers to readily understand and modify the referential constraints.

The needs identified above, and others in which may be set forth below, are satisfied by the invention of this application, which is summarized as follows.

SUMMARY OF THE INVENTION

This invention comprises a computer-implemented, relational data base management system which includes an objective implementation of referential integrity. The system includes at least two relational tables containing records of data, and at least one relationship descriptor. The relationship descriptor describes a referential constraint between the tables, identifying the constraint's parent and dependent tables and primary and foreign keys. The relationship descriptor is a separate object within the data base system and provides the implementation with its objective character. The sys-